ME/R model: A New approach of Data Warehouse Schema Design

[1]Basant Kumar Verma and [2]Sandeep Chaudhary
[1]Associate Professor, IEEE Member, Dept of IT, IET,ALWAR
bitspilani.basant@gmail.com
[2]Assistant Professor, Dept of CS, GEC, Bikaner
kumarsandeep.bits@gmail.com

Abstract: **In this paper we pursue schema design for data warehouses in the spirit of classical database design, organized as a sequence of requirement analysis and specification to collect user requirements, conceptual design to model the data warehouse as a multidimensional database independently from implementation issues, logical design to transform the conceptual data warehouse schema into the target data model, and physical design to maximize performance with respect to a target database system.**

Keywords: DMW, ACID, FDF, ER Diagram

I. Introduction

A data warehouse is an integrated database primarily used in organizational decision making. Although the deployment of data warehouses is current practice in modern information technology landscapes, the methodical schema design for such databases has only been studied cursorily. In short, from a conceptual point of view a data warehouse is a multidimensional database, and *fact schemata*, such as the one shown in Figure 1, represent such databases conceptually.

In Figure 1, we have a fact schema Account from the banking domain, where *measures* Balance, BalanceClass, and No of Transactions are shown in a two dimensional context of *dimensions* Time and Account. Each dimension is specified by means of a lattice of *dimension levels*, whose bottom element is called *terminal* dimension level. Importantly, domains of *optional* dimension levels may contain inapplicable null values, and *context dependencies* specify *contexts of validity* for optional dimension levels (e.g., Age is applicable to private customers, whereas Legal Form is applicable to capital companies, and annotations in the schema indicate these facts). Given a fact schema F, we call the attributes occurring in F the *universe of* F, denoted by UF, and we note that there is a set of functional dependencies over UF , called the *functional dependencies implied by* F, denoted by FDF [1] (e.g., the set of terminal dimension levels functionally determines the set of measures, and each edge in a dimension lattice indicates a functional dependency).

We restrict our attention to the design process up to and including logical design, the relational data model as the target model to be used during logical design, and we emphasize that we do not address design issues that are related to, e.g., cleansing, maintenance, or meta-data management. Our aim is to define a data warehouse design process based on the *extension* of independently developed data warehouse design approaches and their *integration* into the traditional database design process. For this purpose, we strive (1) to identify and formalize desirable warehouse schema properties as design goals and (2) to set up conceptual and logical design phases in such a way that the identified design goals are guaranteed to be fulfilled. Warehouse schemata should satisfy 3MNF, whereas logical data warehouse schemata should be at least update-independent.

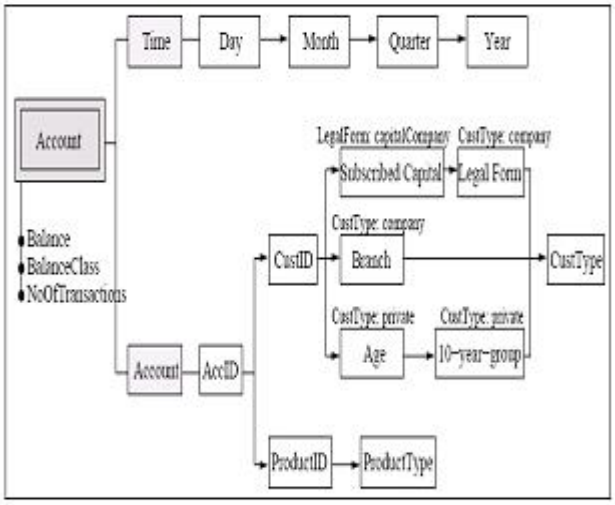

Figure 1. Sample conceptual schema

Traditional database design proceeds in a sequence of conceptual, logical, and physical design steps, where each step results in a corresponding database schema. Importantly, this separation of design phases allows to reason about different aspects of a database system at varying levels of abstraction. There is a growing consensus that this separation of design phases known from traditional database design is also advantageous in the context of data warehouse design. Nevertheless, existing data warehouse design processes lack clearly defined design goals. In particular, previous design processes are unaware of desirable schema properties such as multidimensional normal forms and independence. Indeed, we argue that conceptual data

In this paper we outline a data warehouse design process that comprises the phases of traditional database design and addresses the design goals of normal forms and independence. We assume that data warehouse design starts

DOI: 03.LSCS.2013.4.596

with the conceptual design, which is divided into requirement analysis and design of the conceptual schema; afterwards, logical and physical design are carried out in separate phases. As even a cursory treatment of the physical design phase is beyond the scope of this work, we refer the reader to [2], where an initial set of pointers to relevant research on this subject can be found.

II. Requirement Analysis & Specification

In the first phase of data warehouse design the data requirements to be met by the forthcoming data warehouse have to be analyzed and specified. As opposed to the requirement analysis performed during traditional database design, data warehouse design aims at the integration of a number of pre-existing operational data sources. Hence, the schemata describing these sources form a major input to the requirement analysis. In the course of the requirement analysis, data warehouse designers, warehouse end users, and business domain experts select relevant data warehouse attributes and define initial OLAP queries based on the information found in operational database schemata. We neglect a detailed description of *how* the requirement analysis can be accomplished; instead, we focus on *what* the requirement specification should deliver in order to support the schema design process. To this end, we propose to structure the requirement specification in terms of a set of initial multidimensional or OLAP queries and a derivation and usage table, which describes the identified relevant data warehouse attributes.

The multidimensional queries can be seen as yardstick for the functionality of the data warehouse under design, while the derivation and usage table contains an informal description for each identified warehouse attribute, specifies how its values are derived from the operational databases, and indicates whether the attribute may be used as measure or dimensional attribute. Afterwards, each identified warehouse attribute has to be classified either as measure or as dimension level or as property attribute. Then, to prepare the design goal of multidimensional normal forms, optional dimension levels must be distinguished from mandatory ones, and a context of validity has to be identified for each optional level. In this respect we propose to determine a *basis* for the functional dependencies among warehouse attributes, and we show how such a basis helps to identify (a) measures, (b) certain problems related to dimension levels, and (c) contexts of validity.

III. Conceptual Design

The conceptual design phase performs a transformation of the semi-formal requirement specification into a conceptual schema using a novel, formalized multidimensional data model. The formalization results in a multidimensional diagram such as the one shown in Figure 1, which comprises fact schemata with their related measures and dimension lattices in an intuitive graphical notation. We propose an algorithmic design process to derive fact schemata starting from the requirement specification and functional dependencies of the operational schemata, and we prove that the resulting schemata satisfy 3MNF.As shown in Figure 2, we structure the process of conceptual data warehouse design into three sequential phases: design of initial fact schemata, dimensional lattice design, and definition of summarizability constraints the ME/R model is simple for users to read, and aggregation is shown through the hierarchy of levels representing a dimension.This model resembles the Star model, leading to ease of understanding by both users and logical modelers of the data warehouse.

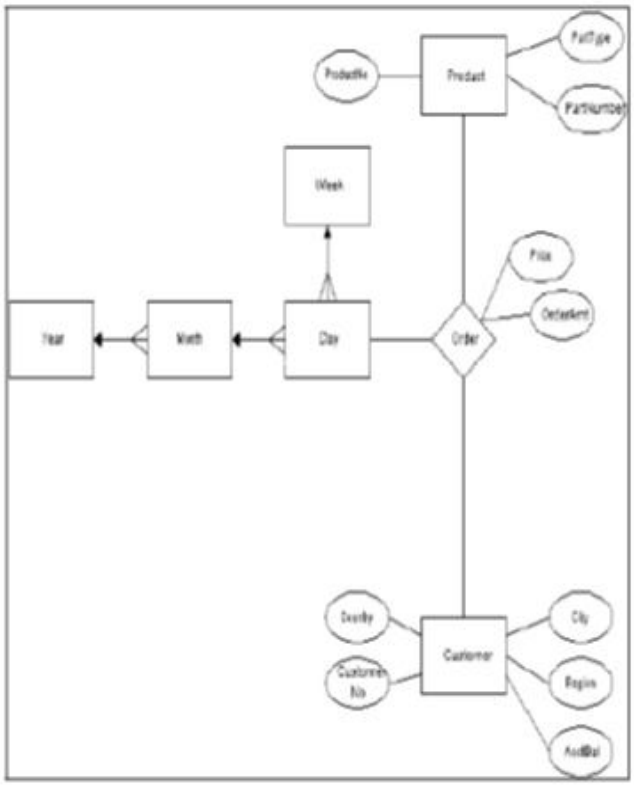

Figure 2: Example fo ME/R Scheme

The definition of initial fact schemata starts from the requirement specification, and the goal is to generate initial fact schemata that contain only terminal dimension levels and measures,thereby specifying the multidimensional context of measures. We just remark that this phase rests upon an analysis of functional dependencies among measures and candidate dimension levels. In the next phase, we gradually develop the dimension lattice for each terminal dimension level. To this end, we augment each initial schema with additional dimension levels by "chasing" functional dependencies starting from the terminal dimension level until no further changes arise. The final phase is about the definition of summarizability constraints. Roughly, all meaningful combinations of measures, dimension levels, and aggregate functions are enumerated.

IV. Logical Design

The conceptual schema as derived according to the previous section is now going to be transformed into a logical

DOI: 03.LSCS.2013.4.596

data warehouse schema. For this purpose, we present a transformation Process that starts from a set of fact schemata in 3MNF and produces a set of view definitions which constitute an update-independent data warehouse. The overall transformation process is sketched in Figure 3. In a first step, a relational database schema DW is generated, which expresses the information modeled by the conceptual fact schemata on a logical level in terms of relation schemata and foreign key constraints. Taking advantage of input fact schemata in 3MNF, null values are avoided in relation schemata representing dimension levels. Afterwards, the resulting database schema is linked to the operational databases. To this end, a set of UPSJR views W over the operational databases is derived such that the materialization of W implements the database schema DW. Finally, the views W thus obtained are rendered update-independent by adding a suitable amount of auxiliary information to W, which ends the logical schema design process. Roughly, we use a so-called view complement [1] as key tool to derive a suitable amount of auxiliary information that renders the data warehouse update independent.

For this purpose, expressions to compute complements for UPSJR views are given. Importantly, these expressions are applicable to a larger class of views than those offered by earlier approaches and lead at the same time to uniformly smaller complements; furthermore, the complexity of constructing these expressions is shown to be polynomial in the size of schema information, which is in striking contrast to previous approaches, which are NP-complete (see also [6]).Besides, we propose a complement-based method to guarantee independence of a set of views with respect to an arbitrary set of operations, which can be applied to enforce independence properties of pre-existing warehouses or to support warehouse evolution (see also [8]).

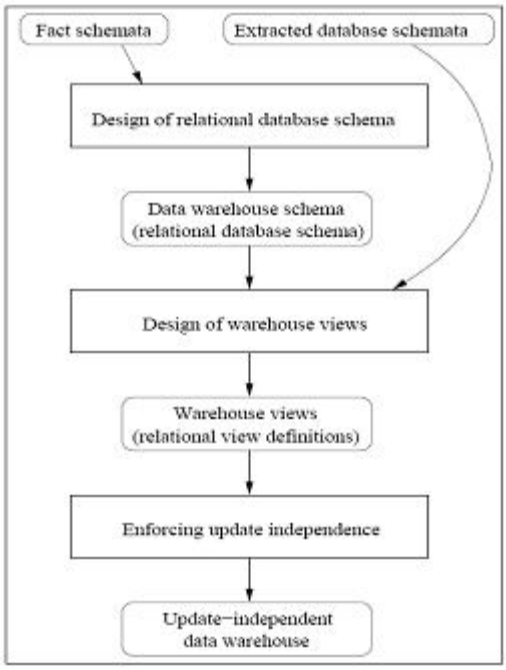

Figure 3. Logical schema design process

V. Conclusion

This work advances data warehouse schema design beyond a process based on experience and rules of thumb towards a formal framework where rigorous methods are available to achieve clearly defined design goals. However, the present approach could be refined and extended in a number of aspects. First, we have neglected physical schema design, which is a mandatory part of any database implementation to optimize the logical schema for performance reasons. Second, meta-data management is largely ignored, and a systematic study concerning the co-design of warehouse schemata and meta-data repository still needs to be performed.

Next, we have not paid attention to special issues that arise when temporal data is stored or analyzed, and the design of data warehouses using temporal data models does not seem to have been studied yet. Finally, we have shown how a data warehouse can be rendered update-independent based on rewritings that add duplicate counters to warehouse views and the additional storage of complementary views to deal with updates concerning join views. Additionally, Proposition 2 suggests that this additional amount of information is even necessary, if join views do not involve projections. In general, however, it remains an open problem to determine a minimal amount of information that is necessary to render a set of views update-independent.

References

[1] F. Bancilhon, N. Spyratos, "Update Semantics of Relational Views," ACM TODS 6,1981, 557–575.

[2] A. Bauer, H. G¨unzel, eds., *Data Warehouse Systeme — Architektur, Entwicklung, Anwendung*, dpunkt.verlag, 2001.

[3] A. Chaudhary, B. K. Verma, J.L. Raheja, "Product Line Development Architectural Model", In proceedings of the 3rd IEEE ICCSIT,China, July, 2010, pp.749-753.

[4] A. Gupta, H.V. Jagadish, I.S. Mumick, "Data Integration using Self-Maintainable Views," Proc. 5th EDBT 1996, LNCS 1057, 140–144.

[5] B. H¨usemann, J. Lechtenb¨orger, G. Vossen, "Conceptual data warehouse modeling,"Proc. DMDW 2000:

[6] D. Laurent, J. Lechtenb¨orger, N. Spyratos, G. Vossen, "Monotonic Complements for Independent Data Warehouses," VLDB Journal 10 (4), Springer-Verlag, 2001, 295–315.

[7] J. Lechtenb¨orger, G. Vossen, "On the Computation of Relational View Complements," Proc. 21st PODS 2002, 142–149.

[8] J. Lechtenb¨orger, G. Vossen, "Multidimensional Normal Forms for Data Warehouse Design," 2002, to appear in Information Systems, Elsevier Science.

[9] W. Lehner, J. Albrecht, H. Wedekind, "Normal Forms for Multidimensional Databases," Proc. 10th SSDBM 1998, 63–72.

[10] H. Lenz, A. Shoshani, "Summarizability in OLAP and statistical databases," Proc. 9th SSDBM 1997, 132–143.

[11] A. Chaudhary, J. L. Raheja, "A Formal Approach for Agent Based Large Concurrent Intelligent Systems", IJAET, Vol. 1, Issue 1, 2010, pp. 95-103.

DOI: 03.LSCS.2013.4.596

[12] S. Kumar, A. Chaudhary, B.K. Verma, "Live Botnet Detection", In proceedings of the IET NCCNS, India, Feb, 2011, pp. 7-14.

[13] A. Chaudhary and et al., "An Algorithmic Approach to Intrusion Detection", In proceedings of the 4th IEEE ICACCT, India, 2010, pp. 742-748.

[14] J.A. Blakeley, P. Larson, F.W. Tompa, "Efficiently Updating Materialized Views," Proc.ACM SIGMOD 1986, 61–71.

[15] S. Kumar, R. Sehgal, P. Singh, A. Chaudhary, "Nepenthes Honeypots based Botnet Detection", JAIT, Vol. 3, issue 4, 2012, pp. 215-221

DOI: 03.LSCS.2013.4.596